เรื่องน่าอ่าน
กิติพงษ์ มุ่งนิรันดร์

บทบาทของ ระบบ เรียลไทม์

เทคโนโลยีที่เพิ่มขีดความสามารถของคอมพิวเตอร์ในการเข้าไปควบคุมงานด้านต่าง ๆ มากยิ่งขึ้น

ระบบเรียลไทม์ (real-time system) คือระบบที่ความสมบูรณ์ของงานไม่ได้ขึ้นอยู่ที่ความถูกต้องของผลลัพธ์เพียงอย่างเดียว แต่ยังต้องประกอบไปด้วยความสามารถที่จะทำให้งานเสร็จออกมาทันเวลาด้วย ระบบเรียลไทม์มีตั้งแต่ระดับไมโครโปรเซสเซอร์จนถึงระบบคอมพิวเตอร์ที่มีขนาดใหญ่และซับซ้อน ระบบต่าง ๆ ในชีวิตประจำวันที่เราต้องเข้าไปเกี่ยวข้องนั้น เราสามารถพบว่าระบบเรียลไทม์ได้เข้ามามีส่วนร่วมอยู่ด้วย เช่น ไมโครคอมพิวเตอร์ควบคุมการทำงานของเครื่องยนต์ ระบบควบคุมการบินในสนามบินได้แก่ระบบ ATC (air traffic control) รวมไปถึงระบบการจองตั๋วเครื่องบิน ระบบทั้งหมดนี้เป็นระบบที่ควบคุมแบบเรียลไทม์

การประยุกต์ใช้ระบบเรียลไทม์ได้เพิ่มจำนวนมากขึ้นและเราจำเป็นต้องเข้าไปเกี่ยวข้องกับมันมากขึ้น ดังนั้นจึงเป็นสิ่งจำเป็นที่เราจะต้องเรียนรู้และเข้าใจเทคโนโลยีด้านนี้ให้มากขึ้น รวมไปถึงการนำเอาเทคโนโลยีนี้ไปใช้ด้วย

ระบบเรียลไทม์

โดยทั่วไประบบเรียลไทม์ประกอบขึ้นด้วยสองส่วน ได้แก่ ระบบควบคุม (controlling system) และระบบที่ถูกควบคุม (controlled system) ตัวอย่างเช่น ในโรงงานอัตโนมัติแห่งหนึ่ง ระบบควบคุมได้แก่คอมพิวเตอร์และระบบติดต่อกับผู้ควบคุมที่ทำการสั่งงานกิจกรรมต่าง ๆ ที่เกิดขึ้นภายในโรงงาน และระบบที่ถูกควบคุมได้แก่ตัวโรงงาน ระบบหุ่นยนต์ สถานีประกอบชิ้นส่วนและส่วนประกอบต่าง ๆ

ระบบควบคุมจะทำการติดต่อกับสิ่งแวดล้อม (environment) หรือระบบที่ถูกควบคุมผ่านทางข้อมูลที่ได้รับจากเซนเซอร์และอินพุตต่าง ๆ สภาพภายนอกที่ระบบคอมพิวเตอร์ติดต่ออยู่

ด้วยเราสามารถมองได้ว่าคือระบบที่ถูกควบคุมนั้นเอง ข้อมูลต่าง ๆ ที่คอมพิวเตอร์ได้รับจำเป็นต้องมีสภาพตรงกับความเป็นจริงของสภาพแวดล้อม หรือมิฉะนั้นก็จะก่อให้เกิดผลเสียหายได้ ดังนั้นการคอยตรวจสอบสภาพแวดล้อมเป็นระยะและการจัดการกับข้อมูลได้ทันเวลาเป็นสิ่งจำเป็นมาก

เวลาเป็นสิ่งสำคัญ

นอกเหนือไปจากความสามารถในการควบคุมให้ระบบทำงานได้ถูกต้องแล้ว ระบบเรียลไทม์ ยังต้องการความแม่นยำด้านเวลาด้วย เช่น ถ้าคอมพิวเตอร์ควบคุมหุ่นยนต์ไม่สามารถสั่งงานหุ่นยนต์ให้หยุดทันเวลาได้ หุ่นยนต์อาจจะชนกับวัตถุอื่นได้ ในระบบเรียลไทม์ส่วนใหญ่ ถ้าเกิดความผิดพลาดไม่ว่าทางด้านตรรกะหรือด้านเวลา ย่อมทำให้งานเสียหายได้

การบังคับด้านเวลา (timing comstraint) สำหรับงานส่วนใหญ่จะแบ่งออกได้เป็นสองแบบ คือ แบบเป็นช่วงหรือระยะ (periodic) หรือแบบไม่เป็นช่วงเวลา (aperiodic) งานแบบเป็นช่วงเวลา หรือคาบจะเกิดขึ้นเป็นระยะ ๆ ที่แน่นอน เช่นเกิดขึ้นทุกเวลา T สำหรับงานแบบไม่เป็นช่วงเวลาจะเกิดขึ้นไม่มีช่วงที่แน่นอน ขึ้นกับเวลาที่เริ่มต้นหรือสิ้นสุดหรือขึ้นกับทั้งสองจุด

งานในระดับล่าง เช่น งานคอยรับข้อมูลจากเซนเซอร์หรืองานคอยส่งสัญญาณไปสวิตช์อุปกรณ์ภายนอกงานเหล่านี้จะมีการบังคับด้านเวลาที่เข้มงวดมาก เนื่องจากคุณสมบัติทางฟิสิกส์ของอุปกรณ์นั้น ๆ เช่นเมื่อต้องการจะอ่านข้อมูลจากเซนเซอร์แต่ละครั้งจะต้องใช้เวลาประมาณ 2-5 มิลลิวินาที ดังนั้นการบังคับด้านเวลาต้องใช้เวลาเพื่ออ่านข้อมูลได้เร็วที่สุดไม่เร็วกว่า 5 มิลลิวินาที การประมวลผลด้านการอ่านข้อมูลส่วนใหญ่จะเป็นแบบช่วง ตัวอย่างเช่น การตรวจอุณหภูมิของแกนเครื่องประติกรนิวเคลียสจะต้องตรวจเป็นระยะ ๆ เสมอเพื่อว่าเมื่อมีการเปลี่ยนแปลงเกิดขึ้นจะได้แก้ไขได้ทันท่วงที

งานนอกจากจะถูกแบ่งด้วยรูปแบบของเวลาแล้ว ยังแบ่งได้ตามสถานการณ์ด้วย ได้แก่ งานที่เกิดขึ้นตั้งแต่ระบบเริ่มต้น (static) และแบบที่เกิดขึ้นชั่วคราว (dynamic) ตัวอย่างเช่น การอ่านอุณหภูมิเป็นงานแบบถาวรตลอดตั้งแต่ต้น สำหรับอีกแบบได้แก่ การที่เครื่องบินบินเข้าสู่สถานีควบคุมการบิน การตรวจสอบเป็นระยะ ๆ จะเกิดขึ้นตั้งแต่เครื่องบินเข้าจนกระทั่งออกจากเขตควบคุม

การบังคับด้านเวลาที่มีความซับซ้อนมากขึ้นอาจมีได้ในระบบ เช่น การพ่นสีรถยนต์ต้องเริ่มหลังจากเวลา T1 และต้องสิ้นสุดก่อนเวลา T2 เหตุการณ์ที่มีวัตถุตกลงด้านหน้าหุ่นยนต์ที่กำลังเคลื่อนที่ หรือการที่ผู้ควบคุมกดปุ่มบนแผงควบคุมซึ่งเป็นงานที่เกิดไม่เป็นช่วงเวลาและเกิดขึ้นโดยไม่คาดคิด (aperiodic + dynamic)

ความต้องการที่มีความสัมพันธ์กับเวลานั้นจะมีผลต่อคุณภาพของงานด้วย คุณภาพของงานที่เสร็จจะเปลี่ยนแปลงตามเวลาที่ใช้ เช่น คำตอบที่ได้ออกมาเร็วและเป็นค่าประมาณอาจจะมีคุณภาพสูงกว่าคำตอบที่ถูกต้องร้อยเปอร์เซ็นต์แต่ช้า หรือถ้างานที่ออกมาเลยกำหนดเวลาไป X ยังยอมรับได้ แต่ถ้าเกินไป X+1 จะยอมรับไม่ได้ เป็นต้น

อะไรจะเกิดขึ้นถ้างานไม่เสร็จตามเวลาที่กำหนด? คำตอบขึ้นกับประเภทของงาน ถ้าเป็นระบบควบคุมโรงไฟฟ้าปรมาณูหรือระบบควบคุมจรวดต้องไม่เกิดข้อผิดพลาดขึ้นในส่วนที่สำคัญ ดังนั้นทรัพยากรหรืออุปกรณ์ที่จำเป็นสำหรับงานที่สำคัญ ๆ จะต้องถูกจัดเตรียมไว้ก่อนแล้วเสมอเพื่อให้ระบบไม่เกิดการล่าช้า แต่อย่างไรก็ตามในบางระบบความผิดพลาดที่เกิดขึ้นอาจยอมรับได้ เช่น การควบคุมหุ่นยนต์ภายในโรงงาน ถ้าเกิดการสั่งงานไม่ทันเวลา ระบบอาจจะหยุดหุ่นยนต์ไว้ก่อน เพื่อรอการสร้างคำสั่งใหม่ขึ้นมาแทน

การเปลี่ยนแปลงที่ต้องยอมรับ

ในระบบเรียลไทม์ที่มีแผนงานที่แน่นอน คุณสมบัติของสิ่งแวดล้อมหรือระบบที่ถูกควบคุมสามารถกล่าวได้ว่าชัดเจนก่อนที่ระบบจะถูกออกแบบขึ้น ดังนั้น รูปแบบและลำดับของงานภายในระบบสามารถที่จะคาดคำนวณว่างานจะเสร็จสิ้นได้เมื่อไร ก่อนระบบจะทำงานจริง ๆ (off line) ระบบแบบ

นี้ไม่มีความยืดหยุ่นแต่ก็ทำให้บรรลุวัตถุประสงค์ของงานได้ทั้งด้านงานและเวลา

ระบบเรียลไทม์ส่วนใหญ่ในปัจจุบันจะมีลักษณะดังกล่าว แต่ระบบในอนาคตจะมีความยืดหยุ่นและเปลี่ยนแปลงมากขึ้น เพราะระบบจะมีความซับซ้อน ระบบจะใหญ่ และกระจายไปตามแต่ละหน่วยงาน (site) และต้องทำงานในสภาพแวดล้อมที่ไม่แน่นอนและซับซ้อนมากขึ้น นอกจากนี้ ระบบจะต้องสามารถถูกปรับปรุงและขยายระบบงานเพิ่มเติมได้ เพราะระบบจะเป็นโปรเจกต์ระยะยาวและพัฒนาตลอดเวลา ดังนั้นระบบเรียลไทม์จำเป็นต้องเร็ว คาดคำนวณได้ เชื่อถือได้ และปรับเปลี่ยนได้

เรียลไทม์ต้องสามารถวิเคราะห์ได้

ความเข้าใจผิดเกี่ยวกับระบบเรียลไทม์ที่ว่าการประมวลผลที่เร็วเป็นความต้องการเพียงอย่างเดียวของระบบนี้ ถึงแม้ว่าความเร็วเป็นคุณสมบัติที่จำเป็น แต่ระบบต้องสามารถเสร็จทันกำหนดเวลาด้วยความเร็ว โดยเฉลี่ยไม่สามารถการันตีว่างานจะทันกำหนดเวลา แต่ถ้าระบบสามารถแสดงได้ว่าเสร็จทันเวลาภายใต้สถานการณ์ที่เลวร้ายที่สุด (worst-case) กล่าวได้ว่าระบบสามารถวิเคราะห์ได้ (predictable)

ความสามารถในการวิเคราะห์หรือคาดเดาได้มีได้หลายความหมาย แต่สำหรับบทความนี้ มันจะหมายความว่า เมื่องานหรือกลุ่มงานได้เริ่มขึ้น มันต้องเป็นไปได้ที่จะตัดสินหรือวิเคราะห์ได้ว่างานจะเสร็จเมื่อเวลาเท่าไร โดยสมมติสถานการณ์ต่าง ๆ ขึ้นด้วย รวมเอาสภาพของระบบ ของระบบปฏิบัติการ (OS) ของทรัพยากรต่าง ๆ ที่ถูกใช้ เข้าไปวิเคราะห์ด้วย

คุณสมบัติของระบบเรียลไทม์

การออกแบบและสร้างระบบเรียลไทม์เป็นงานที่มีระดับตั้งแต่ง่ายจนถึงซับซ้อนมาก ๆ ความยากขึ้นกับคุณสมบัติของระบบ ห้าคุณสมบัติที่สำคัญได้แก่

1. ความเร่งรีบของกำหนดเวลาและเวลาที่ใช้ในงาน

ในระบบเรียลไทม์ บางงานจะมีกำหนดเวลา ถ้าเวลาที่ใช้ระหว่างที่งานเริ่มจนกระทั่งงานสิ้นสุดสั้น จะกล่าวได้ว่างานนั้นต้องเร่งรีบ การทำงานต่าง ๆ ของระบบ ตั้งแต่การตอบสนองของระบบปฏิบัติการรวมไปถึงอัลกอริทึมในด้านการจัดสรรเวลาจะต้องเร็วและง่าย งานที่มีความเร่งรีบสูงย่อมต้องมีความสำคัญมาก ดังนั้นนักออกแบบจะให้ความสนใจในเทคนิคที่เร็วและง่าย เพื่อจัดการกับงานประเภทนี้ ถ้ากำหนดเวลายิ่งเร่งรีบมากขึ้น ความยากของการออกแบบก็จะเพิ่มขึ้นด้วย

2. ความยืดหยุ่นของกำหนดเวลา

รูปแบบของงานจะมีสองประเภท คือ งานที่ไม่มีความยืดหยุ่นเลย (hard) และงานที่มีความยืดหยุ่น (soft) สามารถทำต่อได้ถึงแม้ว่าจะเลยกำหนดมาแล้ว เทคนิคที่ใช้ในงานทั้งสองประเภทนี้จะแตกต่างกัน ในกรณีของงานที่ไม่ยืดหยุ่น งานจะต้องถูกจัดเตรียมและจัดตารางเวลาไว้เพื่อให้แน่ใจได้ว่างานจะเสร็จทันกำหนด สำหรับงานที่ยืดหยุ่นมักจะถูกจัดการด้วยวิธีการที่ใช้ทั่วไปกับงานที่ไม่ใช้เรียลไทม์ โดยเป็นงานที่ต้องปรับให้เหมาะสมระหว่างเวลาที่ใช้กับประสิทธิภาพมากขึ้น งานที่ไม่ยืดหยุ่นจะจัดการได้ยากมากกว่างานที่ยืดหยุ่น

3. ความน่าเชื่อถือ

ระบบเรียลไทม์ที่ต้องการความน่าเชื่อถือสูง เนื่องจากถ้างานที่ถูกเรียกว่างานขั้นวิกฤติ (critical task) ผิดพลาดกำหนดเวลาไป ก็จะก่อให้เกิดความเสียหายร้ายแรงขึ้นได้ งานเหล่านี้จำเป็นต้องการันตีที่จะเสร็จทันกำหนดเวลา โดยผ่านการวิเคราะห์และวิธีการเพื่อสงวนทรัพยากรสำหรับงานขั้นวิกฤตินี้ไว้ให้พร้อมเสมอ ถึงแม้ว่าทรัพยากรเหล่านี้ต้องถูกทิ้งไว้เฉย ๆ เป็นเวลานานก็ตาม

ความต้องการสำหรับงานขั้นวิกฤตินี้คือ งานทั้งหมดต้องเสร็จทันเวลาเสมอ นั่นคือ ต้องไม่มีการผิดพลาดเลย แม้ภายใต้สถานการณ์สมมติต่าง ๆ

4. ขนาดของระบบและระดับของความเกี่ยวข้องกัน

ระบบเรียลไทม์แตกต่างกันไปตามระดับความซับซ้อนและขนาด ในระบบส่วนใหญ่ในปัจจุบันระบบทั้งหมดจะถูกโหลดลงในหน่วยความจำทั้งหมด หรือถ้ามีการแบ่งส่วนออกอย่างชัดเจน

โดยแต่ละส่วนถูกโหลดก่อนการเริ่มใช้งานเสมอ ในงานประยุกต์ส่วนใหญ่ระบบย่อยหรือแต่ละโมดูลจะมีความอิสระจากกัน และมีการติดต่อร่วมมือกันไม่มากนัก ความสามารถในการโหลดระบบทั้งหมดลงหน่วยความจำและการจำกัด การติดต่อภายในระหว่างแต่ละหน่วยให้น้อยที่สุดจะทำให้การออกแบบและวิเคราะห์ระบบเรียลไทม์ทำได้ง่ายขึ้น

แต่สำหรับระบบในยุคหน้าที่ใหญ่และซับซ้อน การที่จะโหลดตัวโปรแกรมทั้งหมดและให้แต่ละส่วนเป็นอิสระต่อกันเป็นจริงไปไม่ได้ นอกจากนั้น การแก้ปัญหาไม่สามารถทำภายใต้ระบบหน่วยความจำเสมือน (virtual memory) ได้เนื่องจากความไม่สามารถคาดเดาถึงเวลาในการอ้างอิงที่แน่นอนลงไปได้ ดังนั้นระบบจะมีขนาดระบบที่ใหญ่ขึ้นและความเกี่ยวข้องกันสูงขึ้นก่อให้เกิดปัญหาใหม่ ๆ เพิ่มขึ้นและทำให้การวิเคราะห์ระบบซับซ้อนขึ้น

5. สภาพแวดล้อม

สภาพแวดล้อมที่เกี่ยวข้องกับระบบเรียลไทม์นั้นมีบทบาทที่สำคัญมากต่อการออกแบบระบบมาก สภาพแวดล้อมส่วนมากจะสามารถรู้ถึงคุณสมบัติที่แน่นอนได้ เช่น เครื่องยนต์รถยนต์ ระบบประกอบเครื่อง เป็นต้น นักออกแบบสามารถกำหนดให้อุปกรณ์เหล่านี้เป็นสภาพแวดล้อมที่วัดได้ (deterministic environment) ถึงแม้ว่าอุปกรณ์เหล่านี้ไม่สามารถวัดได้อย่างแท้จริงแต่เนื่องจากมันสามารถควบคุมได้ สภาพแวดล้อมเหล่านี้ถือได้ว่าเป็นระบบเรียลไทม์ที่คงที่และขนาดเล็ก โดยที่กำหนดเวลาต่าง ๆ สามารถวัดและการันตีได้ตั้งแต่เริ่มระบบ

ปัญหาที่เกิดขึ้นกับระบบขนาดเล็กและการแก้ปัญหานั้น ไม่สามารถนำไปใช้กับระบบที่ใหญ่กว่า ซับซ้อนกว่าและในสภาพแวดล้อมที่ไม่สามารถควบคุมได้ ตัวอย่างเช่น ระบบเรียลไทม์ในอนาคต เช่น กลุ่มทีมงานหุ่นยนต์ขับเคลื่อนที่ทำงานร่วมกันบนดาวอังคาร ระบบนี้จะมีขนาดใหญ่ ซับซ้อน กระจายไปทั่วและปรับเปลี่ยนได้ มันบรรจุรูปแบบต่าง ๆ ของการควบคุมเวลาและการทำงานภายใต้สภาพแวดล้อมที่ไม่สามารถวัดแน่นอนได้และระบบจะทำงานเป็นระยะเวลาที่ยาวนานมากด้วย ระบบที่มีรูปแบบงานที่ไม่คงที่และทำงานภายใต้สภาพที่ไม่แน่นอนเป็นที่ต้องการมากสำหรับการประยุกต์ใช้งาน แต่แนวทางใหม่ ๆ เพื่อให้ได้ระบบที่ปลอดภัยและเชื่อถือได้ยังต้องค้นหากันอีกมาก

ระบบปฏิบัติการเรียลไทม์ (Real-Time Operating Systems)

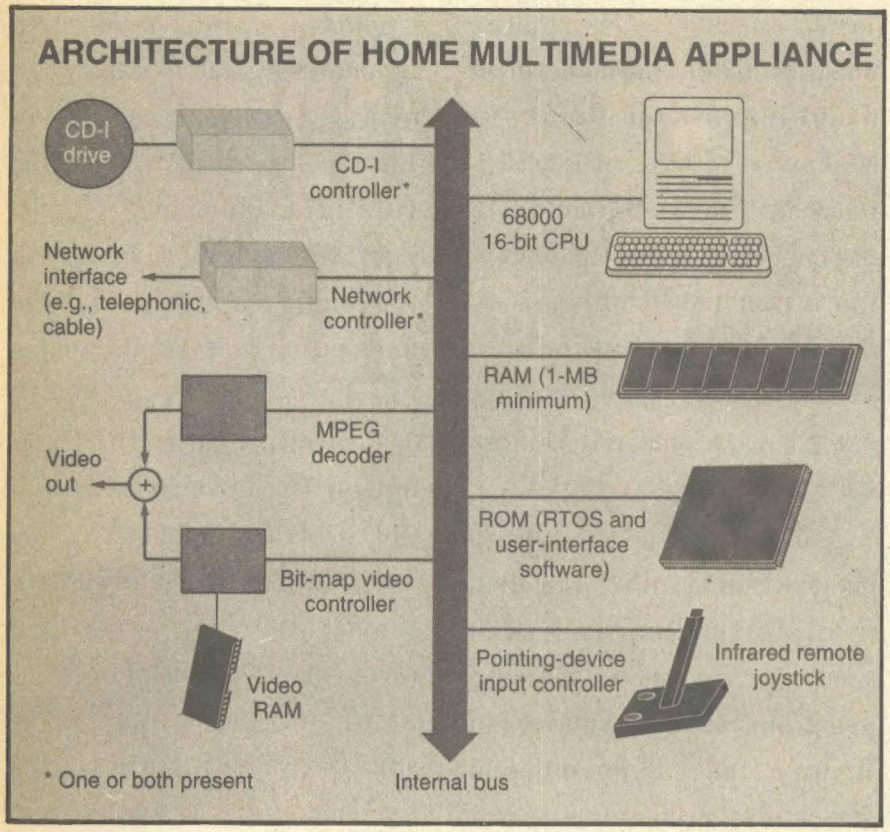
ระบบปฏิบัติการเป็นอีกจุดหนึ่งที่ได้รับการสนใจมากสำหรับระบบเรียลไทม์ในยุคหน้า ระบบโอเอสต้องมีการสนับสนุนขั้นพื้นฐานสำหรับการควบคุมด้านเรียลไทม์ การอดทนต่อความผิดพลาด (fault tolerance) และการกระจายศูนย์ (distribution) รวมไปถึงการจัดสรรเรื่องทรัพยากรสำหรับงานด้านเรียลไทม์และการใช้ร่วมกันระหว่างระบบ รวมไปถึงการจัดสรรระบบสื่อสาร ระบบเซนเซอร์ ซีพียู หน่วยความจำ และระบบอินพุตเอาต์พุตต่าง ๆ

การพัฒนาในอนาคตสำหรับระบบปฏิบัติการเรียลไทม์และแก่นของระบบ (BIOS) ทำให้การออกแบบงาน โดยเฉพาะการสนับสนุนฟังก์ชันพิเศษเพื่อระบบเรียลไทม์ได้แก่ ความสามารถคาดเดาเวลา ความทนทานต่อความผิดพลาดได้ รวมไปถึงความสามารถของระบบโอเอสที่จะสามารถจัดสรรเวลาการใช้งาน ซีพียูและทรัพยากรต่าง ๆ เพื่อให้งานระบบสามารถทำงานได้เสร็จทันกำหนดเวลา

นอกจากนั้น ถ้าระบบโอเอสสามารถตรวจสอบความสำคัญของงานและการตัดสินใจแก้ไขปัญหาที่เกิดขึ้นเมื่อไม่สามารถทำให้งานเสร็จทันกำหนดได้ เช่น การลดประสิทธิภาพของระบบงานลงเพื่อลดความเสียหายให้น้อยลงแทนการหยุดระบบไปเลย

เรียลไทม์กับชีวิตประจำวัน

การรวมตัวกันของคอมพิวเตอร์ ทีวี และโทรศัพท์ จะก่อให้เกิดการบริการและความเพลิดเพลินแบบใหม่ภายในบ้าน ซึ่งเป็นผลมาจากระบบเรียลไทม์ รูปแบบของมันก็คือ การที่มีกล่องเล็ก ๆ ต่อเข้ากับทีวี ประกอบขึ้นด้วยโปรเซสเซอร์ ระบบออดิโอ ระบบ decompression ระบบ signal-processing ระบบ CD และเครือข่าย



รูปที่ 1

ดิจิตอลเน็ตเวอร์กของระบบรวมทีวีและโทรศัพท์ การติดต่อจะผ่านทางระบบรีโมตไร้สายและจอยสติกส์ (Joystick) เพื่อเลือกรายงานหรือสั่งงานต่าง ๆ ผ่านออกไปทางเน็ตเวอร์ก ดังรูปที่ 1

ระบบซอฟต์แวร์ที่ใช้ระบบแบบนี้จะทำงานภายใต้ระบบปฏิบัติการเรียลไทม์ (RTOS) ที่ถูกพัฒนาขึ้นและมีระบบการติดต่อกับผู้ใช้ด้วยระบบกราฟิกที่มีความเป็นกันเองกับผู้ใช้มาก ทุกวันนี้เราสามารถพบซอฟต์แวร์เรียลไทม์ในสินค้าไฮเทคมาก ๆ และในระบบต่าง ๆ ที่ใช้ในอุตสาหกรรม กองทัพ และทางด้านการวิจัยวิทยาศาสตร์ และเมื่อมีการนำเอาระบบมัลติมีเดียมาใช้ร่วมกับซอฟต์แวร์ประเภทนี้ เพื่อใช้ในการติดต่อกับผู้ใช้ ทำให้ระบบดูน่าสนใจมากยิ่งขึ้น

ตัวอย่างแรกของสินค้ายุคใหม่ได้แก่ระบบ CD-I (Compact Disc Interactive) ที่เปิดตัวในเดือนตุลาคม 1991 รูปร่างคล้ายกับเครื่องเล่นซีดี และต่อเข้ากับทีวีคล้ายกับวิดีโอ นอกจากสามารถเล่นระบบเสียงซีดีได้แล้วยังสามารถสร้างระบบภาพและเสียงคุณภาพสูงได้อีกด้วย โดยผู้ใช้สามารถติดต่อและเลือกรายการต่าง ๆ ได้ด้วยระบบติดต่อผ่านทางจอภาพ ตัวอย่างเช่น เด็ก ๆ สามารถดูการ์ตูน สามารถเปลี่ยนสีตัวการ์ตูน เปลี่ยนเสียงซาวด์แทร็ก และสามารถฉายภาพที่แก้ไขแล้วได้ด้วย

ผู้พัฒนาระบบ CD-I ตระหนักถึงความจำเป็นของระบบ RTOS ที่ต้องใช้กับระบบมัลติมีเดียของพวกเขา ดังนั้นจึงได้เลือกระบบโอเอส OS-9 (ระบบโอเอสบนเครื่องแมคอินทอช) เป็นต้นแบบสำหรับระบบ CD-RTOS ที่สร้างเป็นหน่วยความจำถาวร (ROM) ไว้ภายในเครื่องเล่น CD-I เหตุผลที่เลือก OS-9 เพราะว่าความสามารถที่จะปรับเปลี่ยนหรือแก้ไขได้ง่ายของโอเอสตัวนี้ จำเป็นสำหรับระบบอินพุตเอาต์พุตของมัลติมีเดียและการขยายเพิ่มเติมในอนาคต

เรียลไทม์กับระบบมัลติมีเดีย

ข้อมูลสัญญาณภาพและเสียงในระบบดิจิตอลสามารถเก็บและประมวลผลได้เช่นเดียวกับข้อมูลประเภทอื่น ๆ แต่มีข้อแตกต่างที่สำคัญคือ เสียงและภาพจะต้องขึ้นตรงกับเวลา เช่น เมื่อเสียงและภาพถูกฉายซ้ำจะต้องแสดงด้วยอัตราเร็วเดียวกับที่อัดเก็บไว้และเสียงกับภาพต้องแสดงพร้อมกันด้วย เพราะว่าถ้าเสียงและภาพฉายที่อัตราแตกต่างกันเพียง 10 มิลลิวินาที ผลที่ออกมาจะเป็นภาพยนตร์ที่ผิดเพี้ยนไป

ดังนั้น เมื่อต้องการให้ระบบวิดีโอและเสียงเป็นส่วนหนึ่งของโปรแกรมมัลติมีเดีย ซอฟต์แวร์จะต้องสามารถควบคุมและรับรู้ถึงตำแหน่งในการเล่นภาพและเสียง โดยต้องมีความแม่นยำสูงด้วย เพราะถ้าเกิดความล่าช้าในการส่งข้อมูลเสียงไปให้กับระบบเอาต์พุตจะทำให้เกิดเสียงที่ผิดพลาดหรือไม่ต่อเนื่องขึ้นได้

ข้อมูลมัลติมีเดียจะเป็นข้อมูลที่ถูกอ่านขึ้นมาจาก ออปติคัลดิสก์ ระบบสื่อสารข้อมูลความเร็วสูงหรืออุปกรณ์อื่น ๆ ที่สามารถส่งข้อมูลได้ในอัตราเรียลไทม์ สำหรับภาพและเสียงคุณภาพสูงต้องมีอัตราการส่งข้อมูลอย่างต่ำประมาณ 1.5 เมกะบิตต่อวินาที

(Mbps) ดังนั้น โปรเซสเซอร์และระบบปฏิบัติการต้องสามารถจัดการกับข้อมูลมัลติมีเดียพร้อมทั้งทำงานอื่น ๆ ในเวลาเดียวกันด้วย เช่น การรับอินพุตจากอุปกรณ์ที่ต่อพ่วง เช่น คีย์บอร์ด เม้าส์ การแสดงผลออกทางจอภาพ และการทำงานของโปรแกรมอื่น ๆ เป็นต้น

ระบบ RTOS เป็นระบบที่สามารถสนองตอบต่อความต้องการขั้นพื้นฐานสำหรับซอฟต์แวร์มัลติมีเดียได้ เนื่องจากความสามารถด้านมัลติทาสก์ที่ทำให้โปรแกรมสามารถทำงานในเวลาเดียวกันได้เหมาะสมกับงานด้านมัลติมีเดียมาก นอกจากนี้ระบบนี้ยังมีฟังก์ชันการทำงานอื่น ๆ อีก ได้แก่ real-time synchronization, interprocess communications และการจัดการด้านอินเตอร์รัพต์ สำหรับระบบโอเอสที่เป็นที่นิยม ได้แก่ เอ็มเอสดอส และยูนิกซ์ ไม่เหมาะสมกับงานแบบเรียลไทม์เลย

โปรแกรมมากมายที่ทำงานภายใต้ระบบดอสสร้างความปวดหัวให้กับผู้สร้างดอสอีมูเลเตอร์ลงบนระบบโอเอสหรือฮาร์ดแวร์ระบบอื่น ๆ เป็นอย่างมาก เนื่องจากดอสไม่ได้สนับสนุนแนวคิดด้านระบบที่อิสระจากฮาร์ดแวร์ (hardware-independent) ผู้พัฒนาโปรแกรมบนพีซีต้องเขียนซอฟต์แวร์เพื่อสนับสนุนโหมดในการแสดงผลจำนวนมากมายเอง เช่น จอโมโนโครม EGA VGA เป็นต้น ต้องตรวจสอบระบบว่ามีการต่อเม้าส์หรือจอยสติกส์ไว้หรือไม่ ต้องเลือกว่าจะให้สร้างเสียงออกมาทางลำโพงหรือเซาว์บอร์ดไหน ผู้ใช้ต้องปวดหัวในการติดตั้งซอฟต์แวร์ต่าง ๆ เนื่องจากความแตกต่างในแต่ละเครื่อง เห็นได้ชัดว่า ดอสไม่ได้เป็นระบบที่ทำให้ผู้ใช้ใช้งานง่าย ๆ เลย

ดังนั้นระบบ RTOS ที่ดีจะหลีกเลี่ยงปัญหาเหล่านี้ โดยมีคุณสมบัติที่จะขยายระบบได้ง่ายและเร็ว มีระบบอินพุตเอาต์พุตที่ไม่ขึ้นกับฮาร์ดแวร์ ระบบต้องสามารถขยายหรือพัฒนาต่อไปได้โดยไม่มีผลกระทบต่อโปรแกรมใช้งานที่มีอยู่เดิม

สถาปัตยกรรมเรียลไทม์และความทนทานต่อความผิดพลาด

ในอดีตงานเฉพาะด้านที่ต้องการระบบเรียลไทม์จำเป็นจะต้องได้รับการออกแบบระบบ เพื่องานเหล่านี้ขึ้นมาเป็นพิเศษ แต่แนวทางปัจจุบันต้องการให้สถาปัตยกรรมเรียลไทม์มีรูปแบบที่เป็นมาตรฐานมากขึ้น สำหรับบทความนี้จะพิจารณาถึงผลกระทบของสถาปัตยกรรมแบบนี้กับเวลาที่ต้องใช้ในการทำงาน (execution time) ของระบบ และความสัมพันธ์กับ Fault tolerance

เวลาประมวลผลในกรณีที่เลวร้ายที่สุด (worst case) ของโปรแกรมขึ้นอยู่กับฮาร์ดแวร์ ระบบปฏิบัติการ คอมพายเลอร์ และภาษาคอมพิวเตอร์ที่ใช้ คุณสมบัติและเทคโนโลยีทางด้านฮาร์ดแวร์ที่เพิ่มความเร็วเฉลี่ยของโปรแกรมให้สูงขึ้น อาจสร้างปัญหาให้กับระบบเรียลไทม์ เพราะจะทำให้กรณีที่เลวร้ายที่สุดไม่สามารถคาดการณ์เวลาที่แน่นอนได้ เช่น ระบบ caches, pipelining, dynamic RAM และหน่วยความจำเสมือน ทำให้การทำงานของฮาร์ดแวร์ไม่สามารถเดาเวลาที่ต้องใช้แน่นอนได้ การ optimize ของคอมพายเลอร์ทำให้สามารถใช้ฮาร์ดแวร์ได้อย่างมีประสิทธิภาพมากขึ้น แต่ในขณะเดียวกันกลับทำให้เราไม่สามารถวิเคราะห์ถึงเวลาที่ต้องใช้แน่นอนได้ของโปรแกรม นอกจากนี้ยังมีระบบภายในที่ก่อให้เกิดความยุ่งยากในการควบคุมเวลาอีกด้วย เช่น ระบบอินเตอร์รัพต์ การใช้หน่วยความจำร่วมกัน

ระบบเรียลไทม์บางระบบ ประกอบด้วย มัลติโปรเซสเซอร์ เครือข่ายของโปรเซสเซอร์เดี่ยว หรือเครือข่ายของทั้งมัลติโปรเซสเซอร์และโปรเซสเซอร์เดี่ยว สถาปัตยกรรมดังกล่าวมีศักยภาพทาง Fault Tolerance สูง แต่ยากที่จะจัดการให้งานเสร็จตรงตามกำหนดเวลาได้ ดังนั้นในการออกแบบระบบเรียลไทม์ Fault Tolerance ควรจะถูกออกแบบรวมเข้าไปด้วยตั้งแต่ต้น ทั้งทางด้านซอฟต์แวร์และฮาร์ดแวร์ รวมเข้ากับข้อบังคับด้านเวลา Fault Tolerance จำเป็นต้องเป็นระบบที่คงที่ (static task)

การพัฒนาสถาปัตยกรรมและฮาร์ดแวร์เพื่อให้มีคุณสมบัติต่าง ๆ ของระบบเรียลไทม์นั้น ได้มีการทำการวิจัยและพัฒนากันเป็นอย่างมาก เพื่อให้การวิจัยทางด้านนี้มีความก้าวหน้ามากยิ่งขึ้น ได้มีศูนย์สำหรับด้านเรียลไทม์ถูกสร้างขึ้นมาเช่น CRICCS (Center for Real-Time, Intelligent, Complex Computer System) เป็นต้น เราคงต้องมารอดูกันต่อไปว่า เทคโนโลยีด้านนี้จะมีความก้าวหน้าขึ้นอีกแค่ไหน